

Processo de Medição para Micro e Pequenas Empresas baseado em Padrões

Tarciane de Castro Andrade
Grupo de Padrões de Software da UECE
Universidade Estadual do Ceará (UECE)
tarciane@uece.br

Jerffeson Teixeira de Souza
Grupo de Padrões de Software da UECE
Universidade Estadual do Ceará (UECE)
jeff@larces.uece.br

Resumo

Na fase de planejamento do projeto, o gerente de projeto é encarregado de uma das atividades de maior responsabilidade que é a de realizar estimativas quantitativas do software em relação a tamanho, esforço, custo e prazo, como forma de controlar, analisar e melhorar o processo de desenvolvimento de software, além de auxiliar nas tomadas de decisões organizacionais. No contexto das Micro e Pequenas Empresas (MPEs) de desenvolvimento de software, diversas restrições típicas de empresas desse porte, tais como falta de recursos, informalidade excessiva dos processos, imaturidade nas metodologias e crescimento por demanda, tornam o processo de medição ainda mais complexo. Nesse sentido, esse artigo tem como objetivo propor um processo de medição simplificado, integrado ao OpenUP/Basic e baseado em boas práticas de medição documentadas anteriormente como padrões de processo, no sentido de auxiliar MPEs em todas as etapas relativas a esse importante processo.

1. Introdução

Atualmente as empresas têm buscado cada vez mais oferecer ao mercado produtos de software de qualidade. De acordo com Pádua (2005) a existência de processos definidos na organização eleva a sua maturidade, visto que detalha todos os seus aspectos importantes: o que é feito (produto), quando (passos), por quem (papéis), as coisas que usa (insumos) e as que produz (resultados). Com este objetivo, diversos modelos e normas para a definição e melhoria de processos de software têm sido desenvolvidos (SEI, 2005) (KITSON et al., 1996) (ISO, 2006). Porém, estes modelos geralmente são aplicados em empresas de médio a grande porte, pois significam altos custos de implementação, que em geral, não podem ser despendidos por micro ou pequenas empresas.

No âmbito nacional, as Micro e Pequenas Empresas (MPEs) de software são a maioria. Segundo o Ministério de Ciência e Tecnologia (2005), as micro e pequenas empresas são caracterizadas por possuírem até 10 e 50 empregados, respectivamente e, em geral, apresentam uma grande carência na utilização de mecanismos e técnicas para se tornarem mais produtivas e competitivas

no mercado (SERRANO et al., 2006). No Brasil, têm-se percebido um crescimento desta categoria de empresas no mercado, porém, a sua evolução pode ser afetada pela falta de utilização de processos de software que possibilitem o estabelecimento, a padronização e a avaliação das atividades de um projeto de software.

Com a implantação de um programa de melhoria de processos, a organização precisa quantificar sua realidade para elaborar uma avaliação mais objetiva sobre sua evolução. Desta forma, as medições possuem um papel fundamental, pois é através delas que a organização percebe o aumento ou diminuição da qualidade do software produzido.

O presente artigo possui o objetivo de descrever um processo simplificado de medição para MPEs com o intuito de ajudá-las na busca da melhoria contínua.

O público-alvo deste artigo são os integrantes de equipes (gerentes de projetos, gestores da qualidade, analistas, desenvolvedores, testadores, etc) das MPEs que desejam utilizar os avanços da engenharia de software de forma simplificada para aperfeiçoar o desenvolvimento de software, que conseqüentemente, irá proporcionar crescimento e visibilidade à organização.

Esse artigo está organizado da seguinte forma: a seção 2 contém os benefícios e a necessidade de realizar medições, em especial, métricas primitivas como as estimativas, na seção 3 estão descritos os padrões de processos utilizados para compor o processo simplificado de medição que está detalhado na seção 4, na seção 5 estão as conclusões a análise do processo em questão.

2. Medição e Estimativas

Para alcançar níveis cada vez mais altos de qualidade, torna-se necessário melhorar o ciclo de vida do software e, para tornar isto possível, dados quantitativos que descrevam a realidade do processo precisam ser obtidos e devidamente analisados. A medição de software auxilia no monitoramento do processo de desenvolvimento de software, proporcionando sua avaliação contínua e possibilitando ajustes caso seja identificado algum desvio em função das tendências detectadas (ROCHA et al., 2001).

É nesse cenário que a realização das estimativas de software está inserida como um primeiro passo no

processo de medição e garantia da qualidade do processo e do produto. Para Andrade et al. (2004), métricas são metodologias de mensuração cujos principais objetivos, na área de análise de sistemas, são de estimar o tamanho de um software e auxiliar, como indicador, o gerenciamento dos projetos de desenvolvimento de sistemas. Todo projeto deve coletar um conjunto básico de métricas. Este conjunto básico é chamado de métricas primitivas. Cada empresa deve decidir quais métricas primitivas irá coletar. A estimativa de tamanho é uma das métricas primitivas mais utilizadas, pois o tamanho do software tem impacto direto no cálculo de outras métricas.

A estimativa de software é considerada uma das primeiras atividades da fase de planejamento do projeto e é parte essencial da melhoria do processo de software. Estimativas eficientes permitem a verificação da viabilidade do projeto, a elaboração de propostas técnicas e comerciais, a confecção de planos e cronogramas detalhados, o acompanhamento efetivo do projeto (MONTEIRO et al., 2005), controle da produtividade da equipe, do custo, do prazo e do esforço estimado para o desenvolvimento do projeto, alocação adequada da equipe, definição clara das responsabilidades, indicação de performance, avaliação em relação às novas tecnologias, melhoria na precisão das estimativas, entre outros. A partir dos dados estimados é possível controlar o realizado, comparar e manter as medições em uma base histórica e servir de insumo em futuros projetos.

Logo no início do projeto, o gerente de projeto confronta-se com um dilema: produzir estimativas quantitativas, como mensurar custo, tempo e esforço de desenvolvimento do projeto mais próximo à realidade. Para isso, ele deve conhecer a capacidade de sua equipe e os recursos com os quais pode contar para executar as atividades. Desta forma, adequando-se ao custo disponível e à qualidade desejada, o gerente poderá estabelecer prioridades para a realização dessas atividades.

Ao elaborar uma estimativa para o desenvolvimento de um projeto de software é desejável que haja um conhecimento sobre técnicas de estimativas, uma visão global do escopo do projeto a ser gerenciado e um histórico de estimativas de outros projetos semelhantes o que capacita o gerente a quantificar, administrar e planejar mais efetivamente o projeto a ser produzido. Planejar o projeto baseando-se apenas no sentimento ou em experiências anteriores gera, na maioria das vezes, estouro de prazos e elevado custo de desenvolvimento.

Existem atualmente inúmeras técnicas para estimativas de software (ALBRECHT, 1979), (ABRAN et al., 2000), (KARBER, 1993), (IFPUG, 1999), (BOEHM et al., 2000) e (SYMONS, 1991), todas elas na busca constante de realizar estimativas mais próximas do custo real do software.

As empresas de software de uma forma geral buscam estimativas exatas que retratem a eficiência do desenvolvimento e minimizem os fracassos do projeto. As grandes empresas (empresas acima de 100 empregados (MCT, 2005)), em particular, têm facilidade na implantação de técnicas que oferecem maior precisão, em virtude, principalmente, da disponibilidade de recursos humanos, financeiros e possibilidade de investimento em tais técnicas. Para essas empresas, o foco principal é minimizar a margem de erro das estimativas. As técnicas robustas e precisas necessitam de maior esforço para contabilização das medidas, onde na maioria das vezes a medição é feita por fase do projeto, para posterior contagem do todo.

O contexto das MPEs, em contrapartida, possui problemas agravantes e típicos de empresas deste porte, tais como a falta de recursos, a informalidade excessiva dos processos, imaturidade nas metodologias, crescimento por demanda, resistência à inovação e emprego de novas técnicas, falta de atividades de demandas, entre outros. Como consequência, as MPEs, em sua maioria, conseguem apenas se engajar em projetos de pequeno e médio porte.

Segundo ainda o MCT (MCT, 2005), as MPEs enfrentam mais um problema: carência na utilização de métricas consideradas primitivas. Diante destas dificuldades, as MPEs necessitam realizar estimativas de software de forma rápida e simples e que reproduzam significativamente a coleta de métricas, como os valores totais de tamanho, esforço, prazo e custo do desenvolvimento do software (VAZQUEZ, 2003).

Nesse contexto, surge a oportunidade e a necessidade de criação de um processo de medição de estimativas simplificado para as MPEs.

3. Linguagem de Padrões de Estimativa de Software

Padrões são soluções bem sucedidas para um determinado problema (ALEXANDER, 1979) (COPLIEN, 2004). Ao documentarmos um padrão estamos garantindo que ele é largamente utilizado e é considerado uma boa solução para um problema.

Uma linguagem de padrões é uma coleção estruturada de padrões que se apoiam uns nos outros para transformar requisitos e restrições numa arquitetura (COPLIEN, 2004) e é uma forma de subdividir um problema geral e sua solução complexa em problemas menores relacionados e suas respectivas soluções. Uma linguagem de padrões possui um contexto comum compartilhado entre todos os padrões pertencentes a ela. Os padrões podem ser usados isoladamente ou com demais padrões relacionados na linguagem, porém, cada um deles resolve seu próprio problema.

A linguagem de padrões de estimativas apresentada de forma resumida nessa seção foi originalmente proposta em (ANDRADE; SOUZA, 2008). Essa linguagem foi extraída das características comuns das técnicas de estimativas existentes (ALBRECHT, 1979), (ABRAN et al., 2000), (KARBER, 1993), (IFPUG, 1999), (BOEHM et al., 2000), (SYMONS, 1991) e não possui o intuito de estimar o software com a menor margem de erro possível, sendo isto deixado para as diversas técnicas de estimativa de software existentes e que primam por minimizar esta margem. Este trabalho está voltado para oferecer aos gerentes de projetos das MPEs um guia padrão, simples e preciso o suficiente para o processo de obtenção e calibragem das estimativas de software, como forma de atender à realidade dessas empresas e auxiliá-los, juntamente com o cliente, nas tomadas de decisões.

A Tabela 1 é um resumo dos padrões extraídos e que compõem a linguagem software (ANDRADE; SOUZA, 2008).

Tabela 1. Resumo dos Padrões de Estimativas de Software

Nome do Padrão	Problema	Solução
1. Estimar Tamanho	Como estimar o tamanho do projeto?	Para cada funcionalidade classifique-a de acordo com sua complexidade: baixa, média ou alta; Dê pesos e calcule o tamanho total do projeto; Obtenha uma medida de tamanho, chamada de ETT.
2. Estimar Esforço	Como estimar o esforço necessário para execução do projeto?	Obtenha a Produtividade da Equipe (PROD); Obtenha a Estimativa do Tamanho , o tamanho do software (ETT); Estime o esforço total com a fórmula: $ESF = PROD \times ETT$.
3. Estimar Prazo	Como estimar o prazo total de duração do projeto?	Obtenha os valores: Estimativa do Esforço Total (ESF); Obtenção do Valor da Jornada de Trabalho (VlrJTRAB) e Obtenção de Recursos Disponíveis (QtyRD). Estime o prazo total com a fórmula: $EPRZ = \frac{ESF}{(QtyRD \times VlrJTRAB)}$
4. Estimar Custo	Como estimar o valor do custo total do projeto?	Estime com a fórmula, onde ESF é a Estimativa do Esforço Total, P é a porcentagem de esforço necessário para cada papel e VlrHORA é calculado a partir

		da Obtenção do Valor da Hora. $CUSTO = \sum ESF \times PAPEL \times VLRHORA$
--	--	---

4. Processo de Medição de Software

O processo de medição proposto nessa seção tem o intuito de definir as atividades e papéis durante todo o ciclo de vida de desenvolvimento de software e assim contribuir tanto para o aumento do nível de gerenciamento de tais atividades, quanto para inserir, em uma Micro e Pequena Empresa, a base para o processo de medição e controle quantitativo do progresso de desenvolvimento.

O processo de medição foi obtido a partir dos padrões de processos descritos em (ANDRADE; SOUZA, 2008) e resumidos na seção anterior. Ao padronizarmos o processo de desenvolvimento em todas as etapas que envolvem direta ou indiretamente as estimativas de software, contribuimos de forma significativa na coleta de dados mais precisos e conseqüentemente em estimativas futuras mais precisas.

O processo em questão foi elaborado tendo como base o OpenUP/Basic (OPENUP/BASIC, 2006), esse processo possui entre algumas características o fato de unir as vantagens do desenvolvimento iterativo e incremental com as vantagens do desenvolvimento ágil. O objetivo é propor uma extensão do processo OpenUP/Basic com as atividades e marcos de estimativa e coleta das informações como forma de explicitar a importância de tal controle em qualquer processo de desenvolvimento de software. Por limitações de espaço, o presente trabalho descreve apenas as atividades e tarefas pertinentes à medição.

O processo possui os seguintes objetivos:

- Padronizar o levantamento e documentação de requisitos;
- Padronizar o ciclo de vida do caso de uso e o melhor momento para estimá-lo;
- Padronizar o processo de medição baseado nos padrões descritos (ANDRADE; SOUZA, 2008);
- Determinar um processo básico de medição para as Micro e Pequenas Empresas decorrente dos objetivos anteriores e proporcionar a reutilização desse processo padrão;

Os papéis descritos a seguir foram levantados a partir de um estudo em micro e pequenas empresas, onde, por limitação de recursos financeiros, uma única pessoa exerce inúmeros papéis. Para o processo foram utilizados os seguintes papéis:

- **Gerente de Projeto:** é o responsável por planejar, controlar, monitorar todo o processo de desenvolvimento. É ele que realiza as estimativas no início do projeto juntamente com o Analista de

Sistemas. No decorrer do projeto, o gerente compara os dados do planejado com o realizado para tomar possíveis decisões. Ao final do projeto, é o responsável por guardar as informações do andamento do projeto, para servir de insumo para os próximos projetos;

- **Analista de Sistemas:** responsável por coletar e documentar os requisitos do software junto com o cliente. Trabalha junto com o Gerente de Projetos para realizar as estimativas. No decorrer do projeto, informa ao gerente de projetos, o esforço realizado nas atividades;
- **Projetista/Arquiteto:** responsável pela arquitetura e projeto do software. Recebe do Analista de Sistemas a visão do usuário do software e a transforma no projeto (design). No decorrer do projeto, informa ao gerente de projetos, o esforço realizado nas suas atividades;
- **Desenvolvedor:** responsável por desenvolver o software, testar e implantar. Recebe do projetista a visão de design e codifica, testa e implanta o software ou seus módulos. No decorrer do projeto, informa ao gerente de projetos, o esforço realizado nas suas atividades.

O processo OpenUP/Basic é dividido em quatro fases de desenvolvimento, na seguinte ordem: **concepção, elaboração, construção e transição**. Cada fase pode ser planejada por iterações de acordo com a necessidade do projeto. O **Gerente de Projeto**, na fase de **concepção**, entre outras atividades (Figura 1) elabora as estimativas. Para elaborar as estimativas, o Gerente de Projeto necessita que o escopo do projeto esteja bem definido, **Atividade Iniciar Projeto**. Para isso, o **Analista de Sistemas**, juntamente com o **Gerente de Projeto**, elabora o artefato chamado de **Documento de Visão**. O Documento de Visão, nesse momento, possui os requisitos iniciais do projeto e os casos de uso identificados. Posteriormente, a partir de tal documento, o Analista de Sistemas, detalha os casos de uso no documento chamado de **Especificação de Caso de Uso (Atividade Identificar e Refinar os Requisitos)** com os Casos de Uso descritos no formato de *Outline*, com os fluxos básicos e fluxos alternativos identificados.

É o *outline* de caso de uso que servirá de entrada para a atividade de **Planejar Projeto**. Essa atividade é composta de um sub-processo com mais três atividades (Figura 2). Uma das atividades é a **Estimar Escopo**. Entre as tarefas dessa atividade, estão: **Estimar Tamanho, Estimar Esforço, Estimar Custo e Estimar Prazo** (Figura 2). Os cálculos para obtenção dessas estimativas foram descritos e padronizados anteriormente (ANDRADE; SOUZA, 2008). A primeira estimativa é a de tamanho do projeto (Figura 3) que é obtida a partir do **Padrão Estimar Tamanho** (Tabela 1). O padrão Estimar Tamanho foi extraído das técnicas de estimativas

existentes e orientado a caso de uso, o que possibilita a sua utilização de forma genérica e independente de plataforma de desenvolvimento. Uma vez calculado o tamanho do projeto, o próximo passo é o cálculo do esforço, quantidade total de horas para realização do projeto e algumas métricas devem ser levadas em consideração, como: rotatividade de pessoal, produtividade e experiência da equipe.

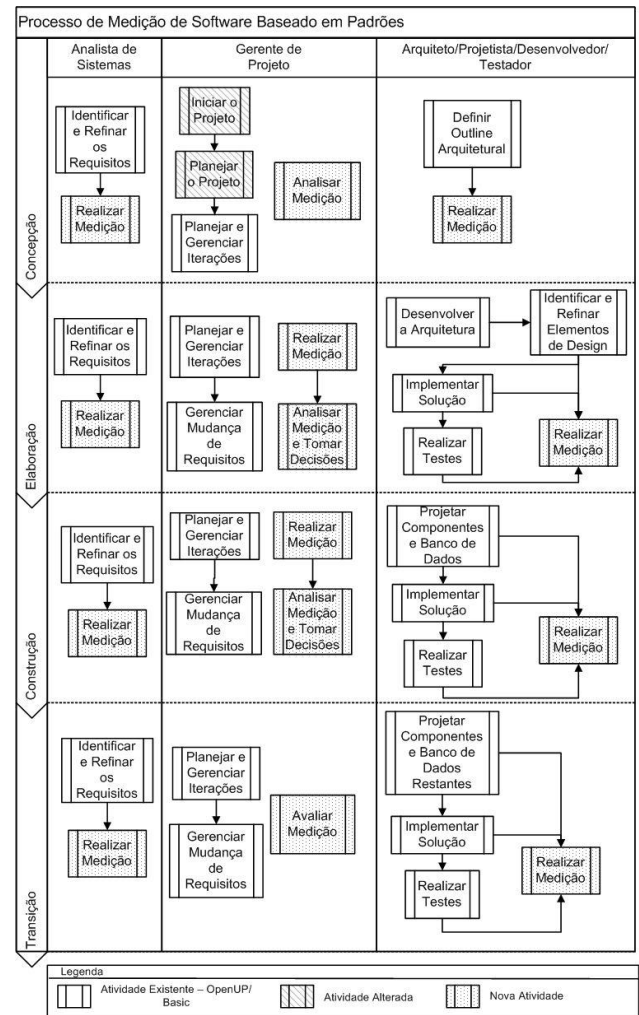


Figura 1. Processo Simplificado de Medição para MPEs

Após o cálculo de todas as estimativas, o passo seguinte é a obtenção de acordo com o cliente sobre os valores estimados.

No decorrer do projeto, ou seja, nas demais fases, os demais envolvidos no projeto (**Analista de Sistemas, Arquiteto/Projetista e Desenvolvedor**) medem o tempo gasto para cada atividade e reportam as horas produzidas para o **Gerente de Projeto**. O Gerente de Projeto de posse das horas realizadas para cada atividade toma as decisões cabíveis para que o projeto mantenha o seu

andamento de acordo com o planejado ou com o mínimo de desvio possível.



Figura 2. Detalhe do Sub-Processo Planejar Projeto

As estimativas produzidas no início do projeto e os dados coletados durante a execução do projeto devem ser analisados e guardados em base histórica para servir de insumo nos projetos subsequentes. Juntamente com os dados, devem ser guardadas informações sobre o projeto tais como: tamanho (classificado em pequeno, médio ou grande de acordo com o porte e perfil da empresa), grau de criticidade, porcentagem de mudança de requisitos (inicial/final) e dificuldades encontradas, por exemplo.

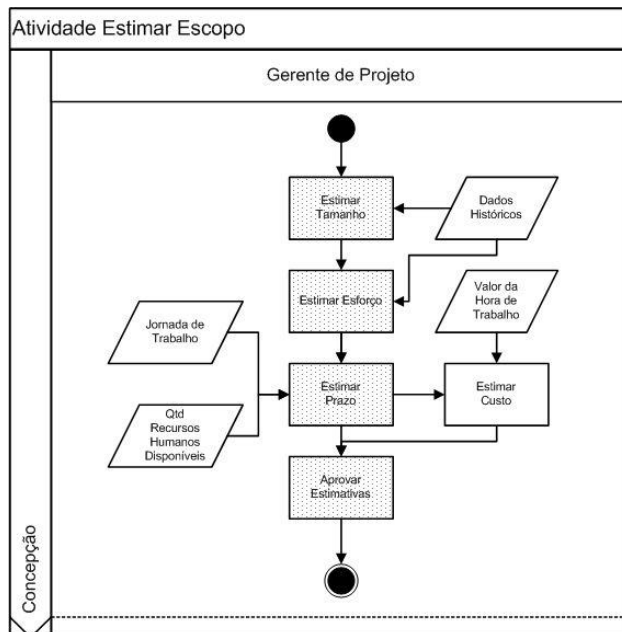


Figura 3. Detalhe da Atividade Estimar Escopo

5. Conclusões e Análise

A comunidade de software vem, nos últimos anos, tentando introduzir nas MPEs a melhoria do processo de software de forma a torná-las mais produtivas e competitivas no mercado e um dos passos importantes para essa melhoria contínua é a medição. Medir o software significa além de obter informações quantitativas, analisá-las e tomar as iniciativas apropriadas para corrigir possíveis desvios durante a execução do projeto, significa também obter dados históricos dos projetos para auxiliar na tomada de decisão futura e na constante melhoria do processo de desenvolvimento.

Existem atualmente inúmeras técnicas de estimar software, algumas simples demais, baseadas na experiência do gerente ou da equipe e outras mais robustas, baseadas de forma mais objetiva na quantidade e qualidade dos requisitos, porém dependem um custo de aprendizado e de aplicação. Os padrões documentados (ANDRADE; SOUZA, 2008) trazem uma forma padronizada de estimar software orientado a caso de uso e independente de plataforma de desenvolvimento.

A partir dos padrões foi construído o processo simplificado de medição de software para as MPEs descrito neste artigo que fornece pontos essenciais para o início do processo de medição nesse nicho de mercado. A estimativa é o primeiro passo para o controle quantitativo do que está sendo produzido e do custo real do projeto para a empresa. Ao estimar no início do projeto e acompanhar o andamento das estimativas no decorrer da execução do projeto, o gerente poderá minimizar os riscos de insucesso do projeto e tomar as medidas necessárias para mitigar possíveis riscos antes que se tornem críticos.

Espera-se como trabalho futuro a implantação do processo de medição em uma pequena empresa de software. Com isso, esperamos consolidar a contribuição com pesquisas relacionadas à área de SPI para MPEs de software, permitindo o amadurecimento dessas empresas no mercado.

5. Referências

- ALEXANDER, C. **The Timeless Way of Building**, Oxford University Press, New York, 1979
- ANDRADE, Edméia; OLIVEIRA, Kátia. **Uso Combinado de Análise de Pontos de Função e Casos de Uso na Gestão de Estimativa de Tamanho de Projetos de Software Orientado a Objetos**. Simpósio Brasileiro de Qualidade de Software, 3., 2004, Brasília.
- ANDRADE, Tarciane; SOUZA, Jerffeson. **Uma Linguagem de Padrões de Estimativa de Software para Micro e Pequenas Empresas**. SugarloafPloP, 2008, Fortaleza.
- ABRAN, A. et al. **Functional size measurement methods: COSMIC-FFP: design and fieldtrials**. FESMA-AEMES Software Measurements Conference, 2000.
- ALBRECHT, A. **Measuring Applications Development Productivity**. *Proceedings of IBM Applic. Dev. Joint SHARE/GUIDE Symposium*, 1979, Monterey, CA.
- BASILE, V; CALDIERA, G; ROMBACH H. **Goal Question Metric Approach Paradigm**. John Wiley & Sons, Encycloppedia of Software Engineering, p. 528-532, 1994.
- BOEHM, B., et al. **Software Cost Estimation with COCOMO II**. Upper Saddle River, NJ, Prentice Hall, 2000.
- COPLIEN, J; HARRISON, N. **Organizational Patterns of Agile Software Development**, Prentice Hall, 2004
- FONTOURA, L., PRICE, R. **Usando GQM para Gerenciar Riscos em Projetos de Software**. Simpósio Brasileiro de Engenharia de Software, Instituto de Informática, 18.,

- Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.
- FRANCA, L., STAAZ, A., LUCENA, C. **Medição de Software para Pequenas Empresas: Uma Solução Baseada na Web.** Simpósio Brasileiro de Engenharia de Software, 7., Rio de Janeiro: PUC, p. 71-84, 1998.
- INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Standard for a Software Quality Metrics Methodology–Description.** Disponível em: <http://standards.ieee.org/reading/ieee/std_public/description/se/1061-1998_desc.html>. Acesso em: julho 2008.
- INTERNATIONAL FUNCTION POINT USERS GROUP (IFPUG). **Function Point Counting Practices Manual.** Version 4.1, January, 1999.
- INTERNATIONAL ORGANIZATION OF STANDARDIZATION (ISO). **International Standard for Software Process Assessment - Part 5: An exemplar Process Assessment Model.** ISO/IEC 15504-5:2006, 2006.
- HAZAN, Claudia. **Implantação de um Processo de Medições de Software, seguindo o Modelo CMMI.** Rio de Janeiro, 2004.
- KARNER, G. **Metrics for Objectory.** 1993. Diploma thesis, University of Linköping, Sweden.
- KITSON, D., et al. **SPICE - Software Process Improvement and Capability Determination.** Disponível em: <<http://www.sqi.gu.edu.au/spice/>>. Acesso em: novembro 2008.
- MAIA, J. **Use Métricas Adequadas – Garanta a Qualidade de Projeto Orientado a Objeto.** Disponível em: <<http://www.euax.com.br/art.00.index.shtml>>. Acesso em: dezembro 2008.
- MINISTÉRIO DE CIÊNCIA E TECNOLOGIA. **Pesquisa Nacional de Qualidade e Produtividade no Setor de Software Brasileiro.** Brasil, 2005.
- MELO, A. **Requisitos de Ferramentas de Apoio aos Processos de Medição de Software.** Relatório Técnico. Belo Horizonte: UFMG, 2007.
- MONTEIRO, Tatiana. **Pontos de Caso de Uso Técnicos – TUCP: Uma extensão do UCP.** Dissertação de Mestrado, Universidade de Fortaleza, 2005.
- OPEN UNIFIED PROCESS BASIC (OpenUp/Basic). Disponível em: <www.eclipse.org/epf>. Acesso em: fevereiro 2009.
- PÁDUA, W. P. **Engenharia de Software – Fundamentos, Métodos e Padrões.** Ed. 2, LTC, 2003.
- PETERS, J; PEDRYCK, W. **Engenharia de Software.** Ed. Campus, 2000.
- PROJECT MANAGEMENT INSTITUTE (PMBOK). **Um Guia do Conjunto de Conhecimentos do Gerenciamento de Projetos.** PMI, 2004.
- PRESSMAN, R. **Engenharia de Software.** Ed. Makron Books, 2006.
- ROCHA, A; MALDONARO, J; WEBER, K. **Qualidade de Software Teoria e Prática.** Prentice Hall, 2001.
- RUIZ, K., FIGUEIRA, F. **Mineração em Métricas de Software.** Escola Regional de Banco de Dados, Rio Grande do Sul: PUCRS, 2007.
- SERRANO, M.; OCA, C.; CEDILHO, K. **An Experience on Implementing the CMMI in a Small Organization Using the Team Software Process.** Proceedings of the First International Research Workshop for Process Improvement in Small Settings – Process Improvement Approaches and Models, Janeiro, 2006.
- SOFTWARE ENGINEERING INSTITUTE (SEI). **CMMI for Development.** Disponível em: <<http://www.sei.cmu.edu/cmmi/>>. Acesso em: 05 abril 2006.
- SOFTEX. **MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Geral.** Julho, 2007.
- SOFTEX. **Guia de Implementação Parte 2 – Nível F.** Julho, 2007.
- SYMONS, C.R. **Software Sizing and Estimating, MKII FPA.** John Wiley and Sons, 1991.
- VASQUEZ, C. E. **Análise de ponto de função: medição, estimativas e gerenciamento de projetos de software.** 1 Ed. São Paulo: Érica, 2003.